DIRNAME

Vulnerable to TOCTOU issues

Sean Barnum, Cigital, Inc. [vita[1]]

2007-03-22

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 7279 bytes

Attack Category	• Path spoofing or confusion problem
Vulnerability Category	• TOCTOU - Time of Check, Time of Use • Indeterminate File/Path
Software Context	• Filename Management • File Path Management
Location	• libgen.h
Description	Note: dirname, basename functions should be analyzed together The dirname() function takes a pointer to a character string that contains a pathname and returns a pointer to a string that is a pathname of the parent directory of that file. Trailing '/' characters in the path are not counted as part of the path. If the path does not contain a '/', then dirname() returns a pointer to the string "." . If the path is a NULL pointer or points to an empty string, dirname() returns a pointer to the string "." . A call to dirname() should be flagged if the argument (the directory name) is used previously in a check-category call.
APIs	**Function Name** / **Comments** dirname / use basename / (for reference; See basename rule)
Method of Attack	The key issue with respect to TOCTOU vulnerabilities is that programs make assumptions about atomicity of actions. It is assumed that checking the state or identity of a targeted resource followed by an action on that resource is all one action. In reality, there is a period of time between the check and the use that allows either an attacker to intentionally or another interleaved process or thread to unintentionally change the state of the targeted resource and yield unexpected and undesired results.

1. http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html (Barnum, Sean)

The dirname() call is a use-category call, which when preceded by a check-category call can be indicative of a TOCTOU vulnerability.

A TOCTOU attack in regards to dirname() can occur when

a. A check for the existence of a file or other reference to a file/directory name, to be parsed by dirname, occurs

b. The dirname call is executed to return the pathname of the parent directory of the directory/ filename.

Between a and b, an attacker could, for example, link the target file (the file to be parsed) to a different known file. The subsequent parse of the target file would result in minimally, potentially erroneous program function.

Exception Criteria

Solutions

Solution Applicability	Solution Description	Solution Efficacy
Generally applies to any dirname().	Translate dirname() into function(s) using file descriptors, if possible.	See notes
Generally applies to any dirname().	The most basic advice for TOCTOU vulnerabilities is to not perform a check before the use. This does not resolve the underlying issue of the execution of a function on a resource whose state and identity cannot be assured, but it does help to limit the false sense of security given by the check.	Does not resolve the underlying vulnerability but limits the false sense of security given by the check.

	Generally applies to any dirname().	Limit the interleaving of operations on files from multiple processes.	Does not eliminate the underlying vulnerability but can help make it more difficult to exploit.
	Generally applies to any dirname().	Limit the spread of time (cycles) between the check and use of a resource.	Does not eliminate the underlying vulnerability but can help make it more difficult to exploit.
	Generally applies to any dirname().	Attempt the dirname and then perform an application-specific sanity check after the dirname call.	Effective in some cases.

Signature Details	char *dirname(char *path);
Examples of Incorrect Code	``` int main() { struct stat stats; char *path="/etc/passwd"; char *dname; stat(path, &stats); dname=dirname(path); return 0; } ```
Examples of Corrected Code	``` int main() { struct stat stats; char *path="/etc/passwd"; char *dname; dname=dirname(path); /* An application-specific sanity check needs to occur at this point to verify expected behavior occurred*/ return 0; } ```
Source References	• Viega, John & McGraw, Gary. *Building Secure Software: How to Avoid Security Problems*

	the Right Way. Boston, MA: Addison-Wesley Professional, 2001, ISBN: 020172152X, ch 9 • Unix man page dirname()
Recommended Resource	
Discriminant Set	**Operating System**: • UNIX **Languages**: • C • C++

Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. mailto:copyright@cigital.com